# MySTEMGrowth Survey

## sdmay26-27

# Meet the team

**Caleb Hemmestad**
Software Engineer
Backend/Cloud

**Nina Gadelha**
Cybersecurity Engineer
Cybersecurity Developer

**Ethan Buenting**
Computer Engineer
Backend/Cloud

# Meet the team

**Ethan Van Caster**
Computer Engineer
Frontend, Client Contact

**Sam Craft**
Software Engineer
Backend Developer

**Ryan Mamrot**
Software Engineer
Frontend Developer

# Presentation Overview

# Presentation Overview

# Presentation Overview

# Project Problem

| Why was the current survey not ideal? | What were the desired improvements to the existing tool? | What trade offs were there with a standalone tool? |
| --- | --- | --- |
| → Data Collection was a manual process<br>→ Returning results to students was delayed<br>→ No way to provide extra resources | → Return results immediately to students<br>→ Show differential between survey attempts<br>→ Provide links to extra resources | → Tool had to be built from scratch<br>→ Hosting the website would have to be part of the project<br>→ No bug/issue fixing by third party, all on the team |

# Project Introduction

## What is the MySTEMGrowth Survey tool?

- ➔ Website + survey tool
- ➔ Supports 3 types of users
- ➔ Program affiliated Institutions

## How can the MySTEMGrowth Survey be used

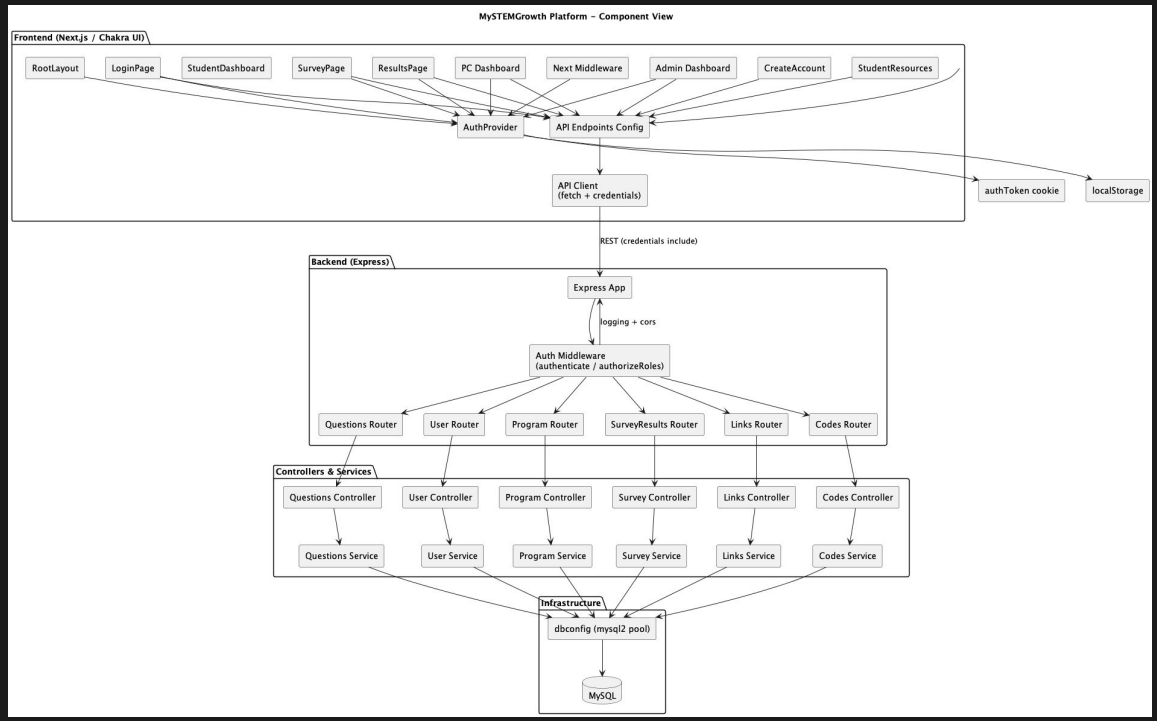
- ➔ Take the survey
- ➔ Complete undergraduate research program
- ➔ After course completion, take the survey again to compare results

## What results does the MySTEMGrowth Survey provide?

- ➔ Civic Engagement
- ➔ STEM Interests
- ➔ STEM Self Efficacy
- ➔ STEM Outcome Expectations
- ➔ Research Outcome Expectations
- ➔ Research Self Efficacy

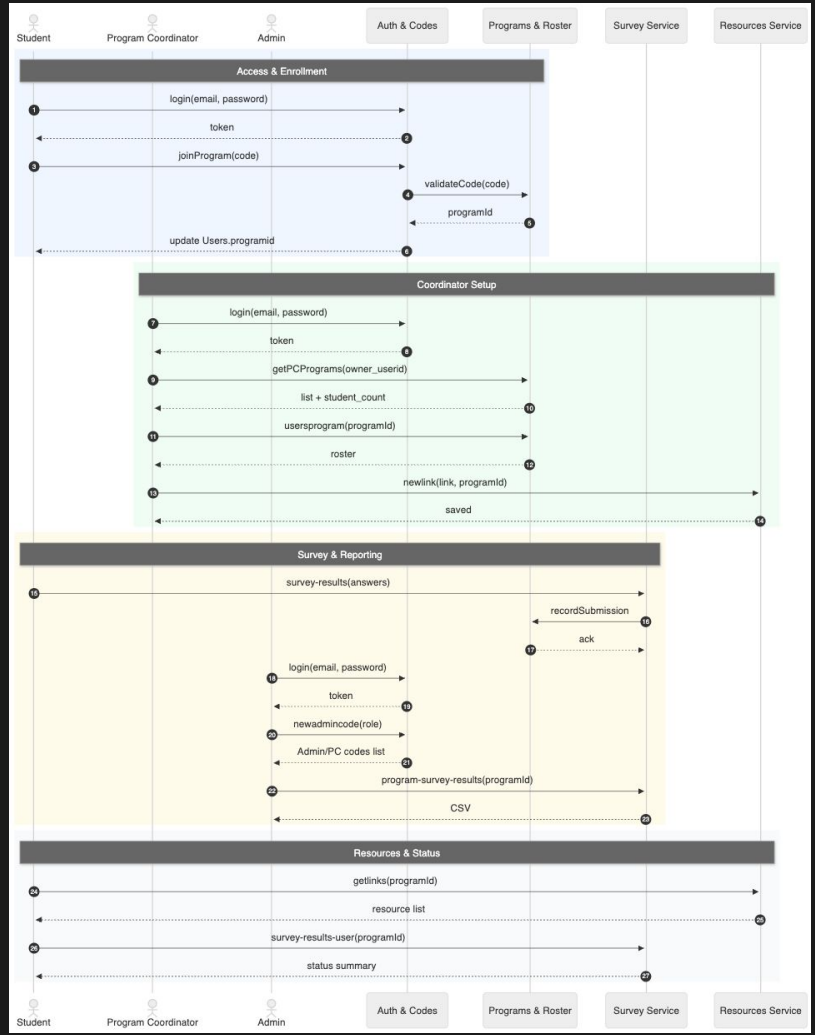
# Full Stack Overview

- The Frontend is built with React.js and Node.js to build the interactive elements that we have discussed so far
- Node.js is used to build the backend and controller layer. Routers are built to be called by the web app, and use the controllers and service to interact with the cloud infrastructure.
- DBconfig is a JavaScript framework called by all the Service systems that allows them to interact with the database. It inherits the database from Environment Variables upon launch of the program.



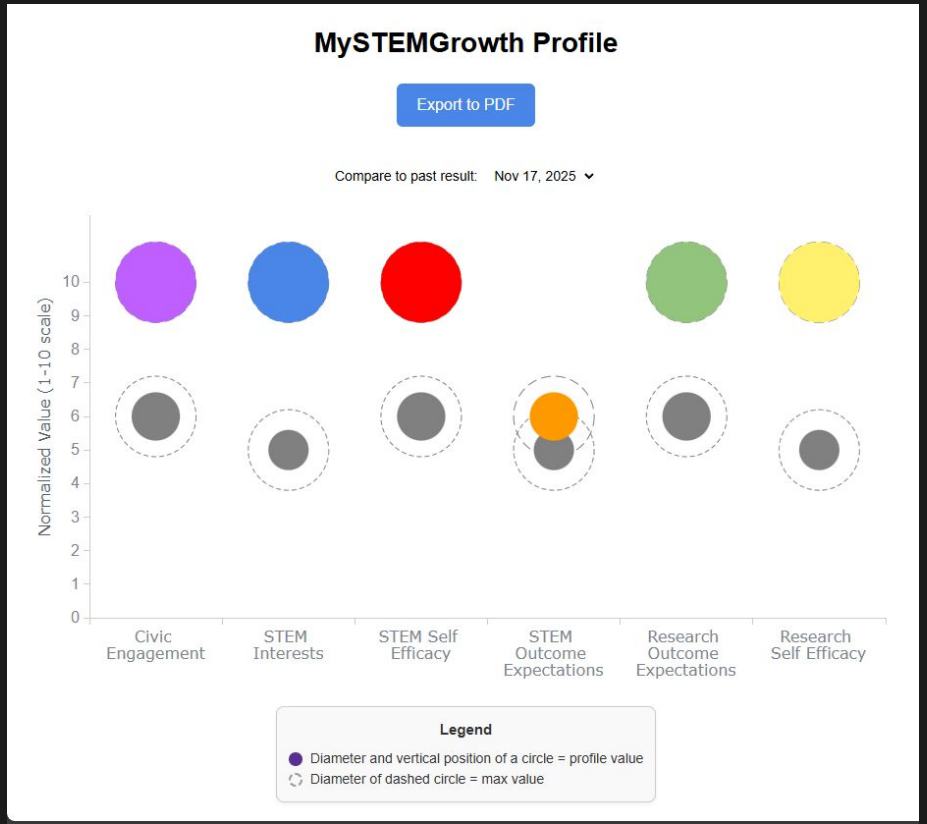
# User Interaction & Timeline

- ➔ Students
  - Simple enrollment and secure login flow
  - Clear access to assigned surveys
  - Minimal effort to submit responses
  - Confirmation that submissions are recorded
- ➔ Program Coordinators
  - Efficient program setup and student onboarding
  - Add or delete resources for programs
  - Ability to manage rosters
- ➔ Administrators
  - Create and manage programs and coordinators
  - Control access codes and permissions
  - Monitor system-wide survey activity
  - Ensure data integrity and consistency
  - Export survey results for reporting and evaluation



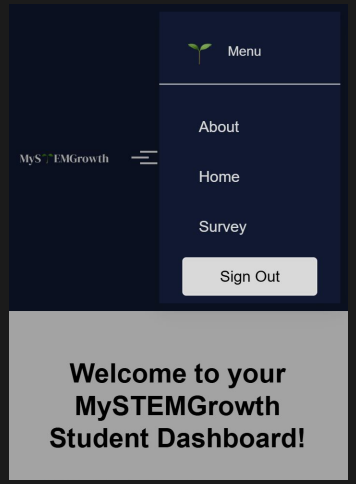
# Student User Overview

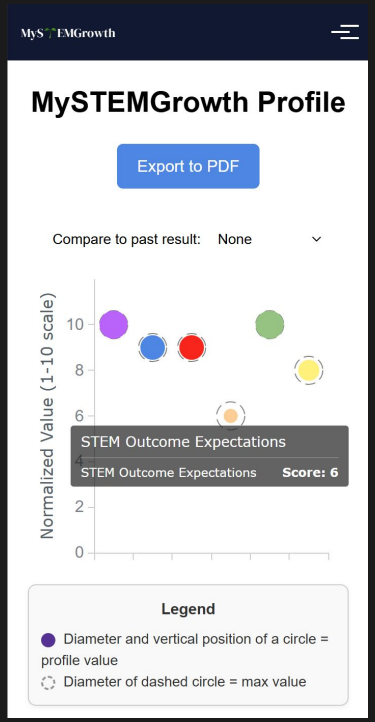
- A dashboard was created to give users easy access to the survey, results and additional resources
- After the survey is taken, results are immediately available for students to see
- Results are able to be viewed comparatively by a dropdown, allowing students to see what areas they have improved in for their program

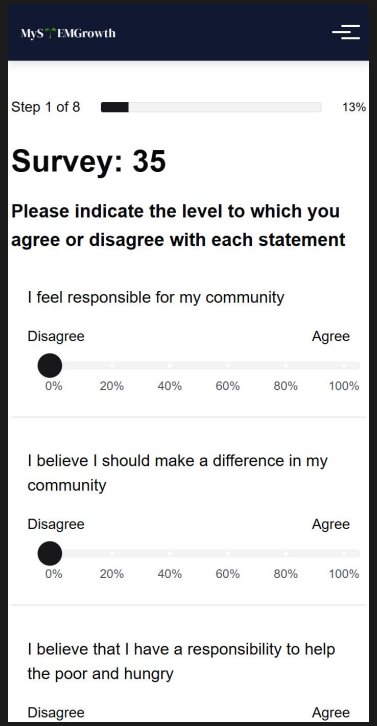


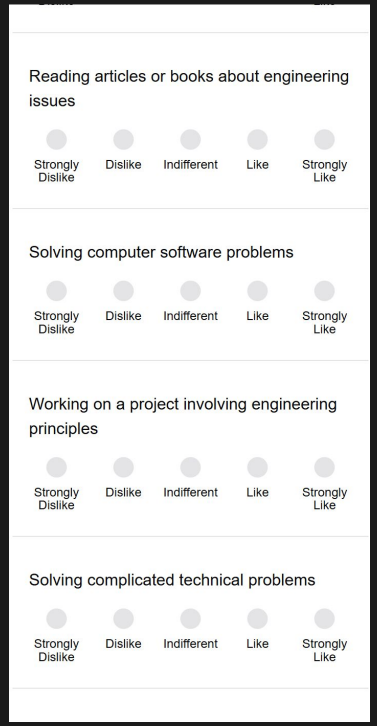
# Mobile View

- One of the most prominent features that this app was lacking when we inherited the project was mobile compatibility. Survey had to be taken on a computer web browser to be usable



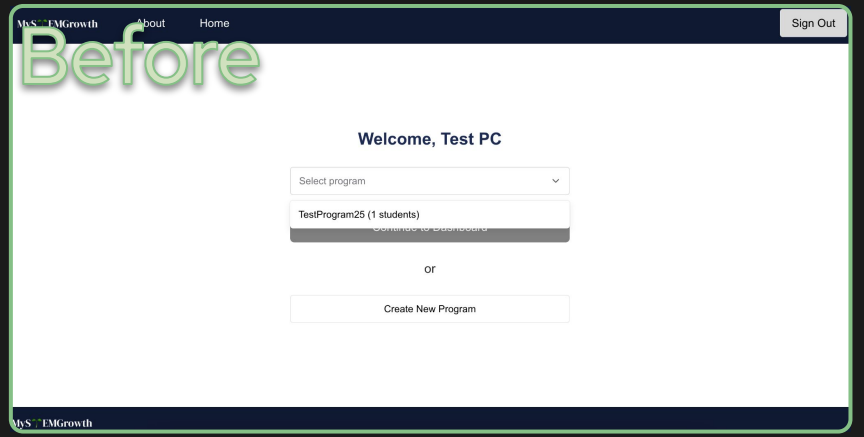



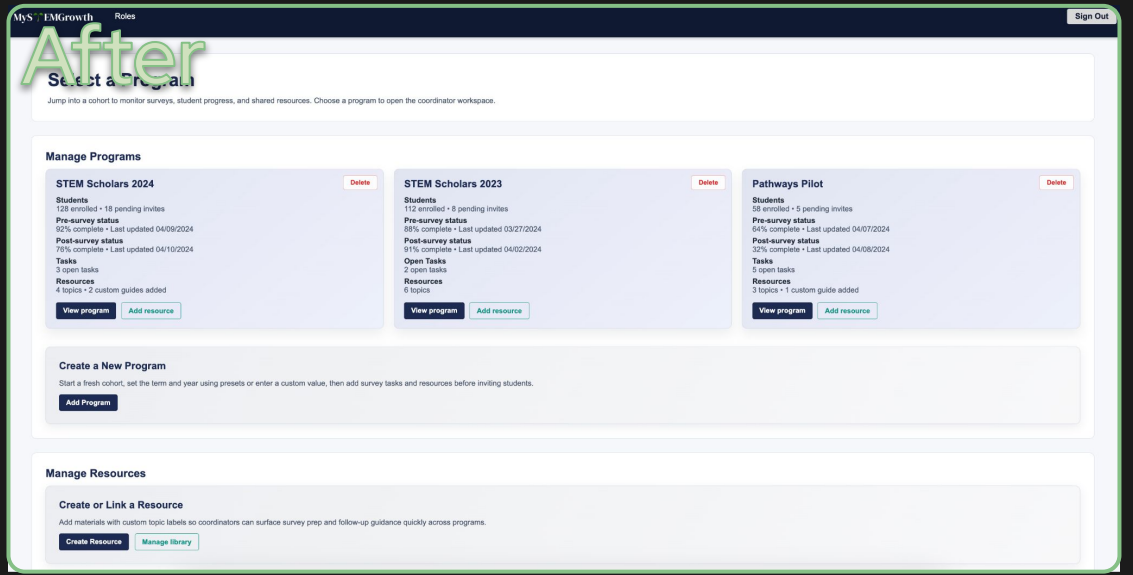



# Program Coordinator Overview

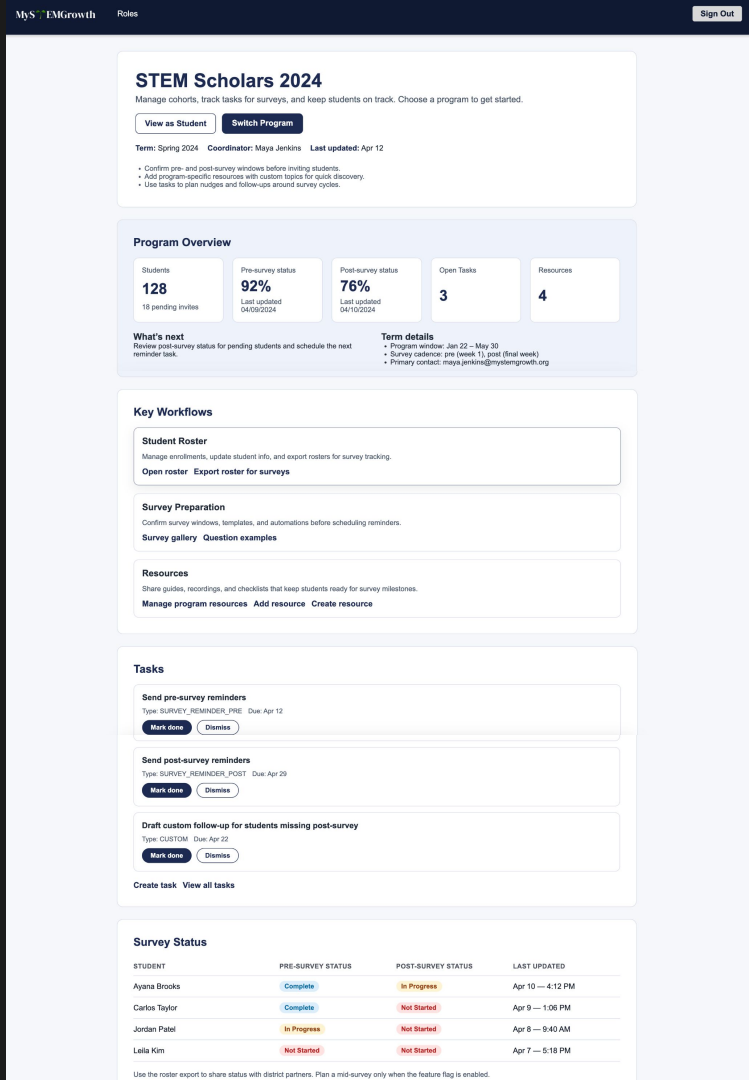
- The program coordinator creates programs and manages them by inviting students, starting surveys, and adding resources
- They can manage multiple programs but only have access to their own programs
- Functionally, program coordinator operates the same with adjustments to the User Interface for a better overview of current programs and easier/more straightforward navigation



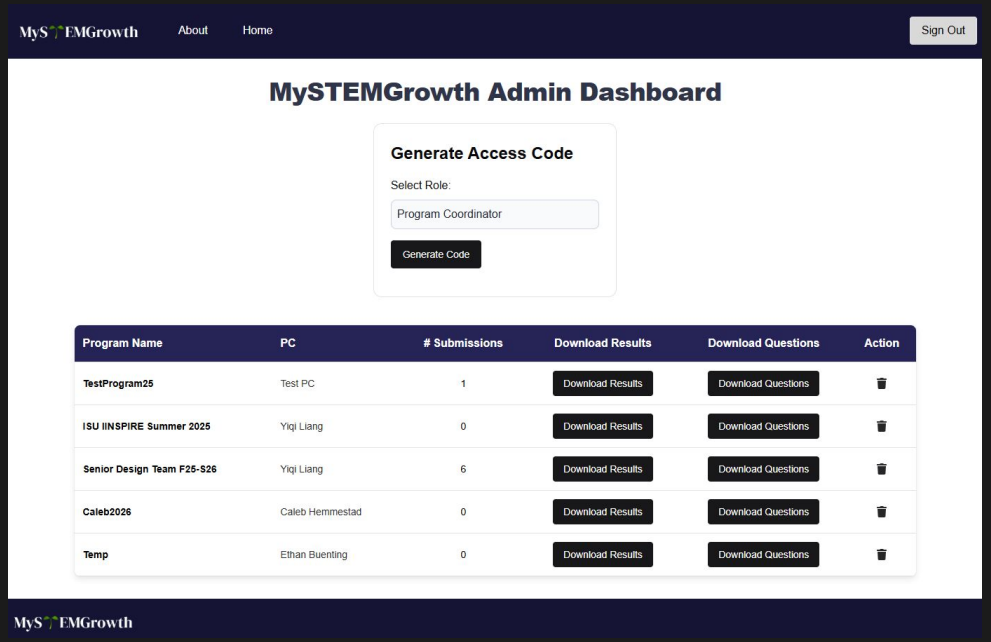
# Preliminary Program Coordinator Improvements

- With the new centralized program coordinator dashboard, it makes getting information about the progress of programs much easier and reduces time spent navigating between programs and tools
- Key workflows, like student roster and resources, are easy to get to so the program coordinator can get the information that they want quickly without having to search for it
- Tasks help the program coordinator stay organized and on track with everything that's going on so that they don't forget crucial actions involved with that program, making managing multiple programs easier
- At-a-Glance Survey Status with visual indicators for completed, in-progress, and missing responses helps identify students to follow up with quickly



# Admin Overview

- ➔ Can export survey data and questions from any program
- ➔ Generates codes to create Program Coordinator and other Admin users
- ➔ Able to modify the survey: add, edit or delete questions



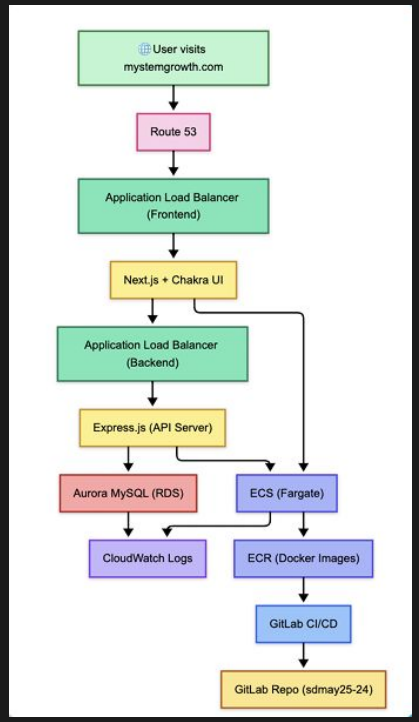
# Cloud Infrastructure

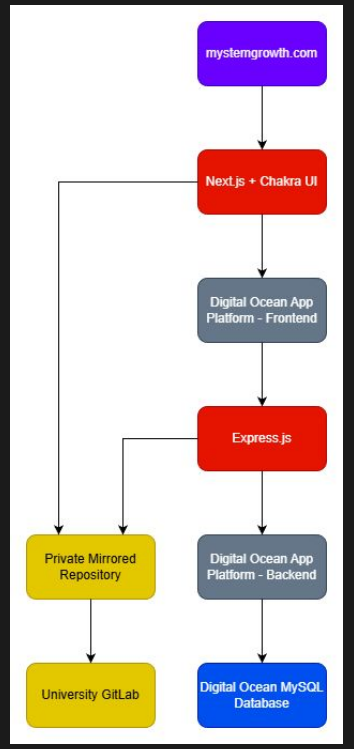
- Previous Implementation of the application used Amazon Web Service
- While the hosting service worked, the cost was high and implementation was complicated do to the additional steps needed
- After looking at multiple options, the team opted to move cloud infrastructure to Digital Ocean

| CRITERIA | WEIGHT | AWS | DIGITALOCEAN | GCP | AZURE | HEROKU/VERCEL |
|---|---|---|---|---|---|---|
| Cost | 0.25 | 2 | 5 | 3 | 3 | 4 |
| Ease of Setup | 0.20 | 2 | 5 | 3 | 3 | 5 |
| Integration w/ Stack | 0.20 | 5 | 5 | 4 | 3 | 3 |
| Security | 0.20 | 5 | 4 | 5 | 5 | 3 |
| Scalability | 0.15 | 5 | 4 | 5 | 5 | 3 |
| Weighted Score | | 3.7 | 4.7 | 4.2 | 4.0 | 3.8 |

# Digital Ocean

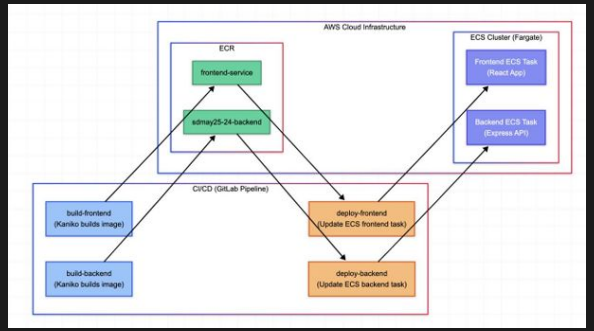
- ➔ This platform was over 10x cheaper a month to host, with identical performance on the user side of the platform
- ➔ Allowed for a more simplistic implementation, made up of a singular App Platform and Database
- ➔ App Platform has 2 components, 1 for frontend and 1 for backend and controller handling
- ➔ This design allows future admins and maintainers to easily scale the application;s resources should the need arise without needing to learns the ins and out of the platform

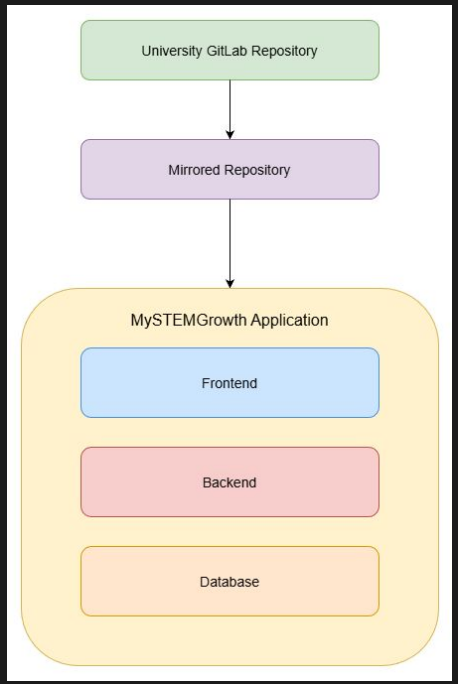


# CI/CD Changes

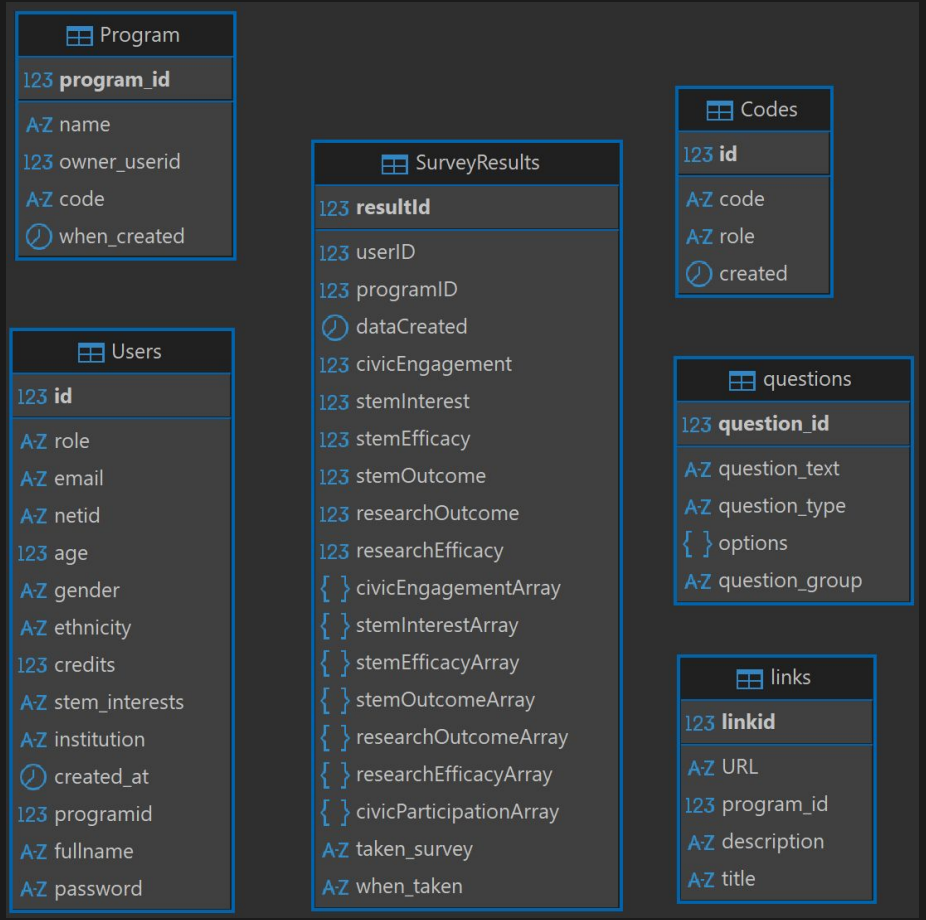
- ➔ Digital Ocean offers less flexibility than Amazon Web Service on how it deploys applications
- ➔ Could not upload directly from the University Git, so a mirrored repository was needed
- ➔ This solution maintains the same level of automation



# Backend

- ➔ Given a database with data already in it, had to make sure all data was preserved no matter what was done
- ➔ Small errors in questions table
- ➔ Backend works for what tool was, needs work for desired improvements
- ➔ Most tables act independently when tool is being used



Basic Security Testing

# You Can’t Patch What You Haven’t Poked!

### Vulnerability Scans

→ Outdated software

→ Server misconfigurations

→ Known vulnerabilities

### File Enumeration

→ Unauthorized file access

→ Sensitive file display

→ Directory listing enabled

### Authentication Testing

→ Login protections

→ Session handling

→ Error messaging

### Input Validation

→ SQL injection

→ Cross-site scripting

→ Command injections

### Password Security

→ Password storage

→ Password practices

→ Password reset

Vulnerabilities found in website's past deployment

# And the Results are.....
(not great)

## Information Leaks

Sensitive information was exposed for anyone to access

- ➔ Passwords stored in plaintext
- ➔ Unprotected endpoints
- ➔ Sensitive files exposed

## Absence of Middleware

Authentication was not supported

- ➔ No cookie implementation
- ➔ Lack of tokens
- ➔ Little authentication required

## Password Handling

Secure password practices were not implemented

- ➔ No password hashing
- ➔ Weak password practices
- ➔ Unimplemented  account protections

Security Innovation Plan

# Improvements to Implement

1

Design & Implement MIddleware

→ Tokens
→ Cookies

2

Enforce Secure Password Practices

→ Password hashing
→ Require strong passwords
→ Add login protections

3

Protect Sensitive Endpoints & Files

→ Restrict access to website files
→ Limit endpoint access

4

Integrate Robust Input Validation & Injection Protection

→ Ensure proper input handling
→ Prevent injection attacks

# Planning for the future

## Improving User Dashboard

- User dashboard is sparse, could clean up the overall look
- Create task cards, similar to canvas for individual programs
- Add an option for users to add more programs later, not just at account creation

## Updating Export Data Option

- Program Coordinators and Admins can both export data now
- Improve the exportation function to align more closely with researcher's requested format
- Add export to .sav SPSS file

## Admin's Program Search

- Admins currently just see a list of all programs on the application
- To ensure the app is usable as it grows, adding search and filter options for programs is needed

## Survey Improvements

- Double check the calculations of the survey are working as expected
- Add support for a partial survey being saved for later completion
- Add an option for Program Coordinators to collect identifying information if it aligns with their needs and is within regulations

# Questions?